Attorney's Docket No.: 10559-325001 / P9686

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: ADAPTIVE READ PRE-FETCH

APPLICANT: GARY A. SOLOMON

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No EL624272781

I hereby certify under 37 CFR §1 10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D C 20231

March 27, 2001
Date of Deposit

Samantha Bell
Signature

Samantha Bell
Typed or Printed Name of Person Signing Certificate

Adaptive Read Pre-Fetch

FIELD OF THE INVENTION

This invention relates to computer read pre-fetch operations,.

BACKGROUND OF THE INVENTION

. Existing PCI bridges assist in the control of the sequencing of operations and access to computer busses in accordance with the bus specification (such as, for example, PCI Local Bus Specification Rev. 2.2 published by the PCI Special Interest Group). Pre-fetch algorithms are not covered by the PCI specification, but are widely employed by PCI devices to circumvent a fundamental issue with PCI protocol: it does not include a read amount embedded within each transaction. Such devices employ a static read pre-fetch which requests the same amount of information for a particular type of read operation, regardless of the actual demands of the requesting agent. While this constant pre-fetch amount may be adjustable by means of a device specific configuration register, the selected amount is constant and applicable to all requesting agents served in connection with that register. A static pre-fetch amount may result in pre-fetching too much data.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of an adaptive read pre-fetch system.

FIG 2. is a flow chart of an adaptive pre-fetch read method.

DETAILED DESCRIPTION

Referring to FIG 1, an example adaptive adaptive read pre-fetch system 10 is shown having components on a bridge 12. The components include a pre-fetch factor register 15, being a re-writeable storage location. The adaptive read pre-fetch system 10 also includes a re-read pre-fetch factor register 20, a re-read timer 25 and a next read address register 30. Also shown is pre-fetchable data storage such as system memory 40, and agents 35a, 35b and 35c. Each of the components of the adaptive read pre-fetch system 10 are preferably part of or attached to the computer, such as a bridge 12, within which the pre-fetch factor register 15, the re-read pre-fetch factor register 20, the re-read timer 25, and the next read address register 30 may, but need not, reside. Also shown in Fig. 1 is a CPU 42 which communicates through a host bridge 44 with a PCI primary bus 46. Bridge 12 is also capable of communicating with the primary bus 46.

An agent 35a, 35b or 35c may be any requesting agent, such as an agent on a PCI 2.2 secondary bus 30 connected to a bridge 12. An agent may be any of a number of devices capable of requesting a memory read operation on the bus.

At a set time, typically upon system reset, the values in the pre-fetch factor register 15 and re-read pre-fetch factor register 20 are initialized.

When an agent on the bus 30 requests a memory read operation, it notifies bridge 12 of the request by asserting the appropriate signals on the bus 30. If the bridge 12 determines that the request is from pre-fetchable storage 40, it multiplies a pre-defined amount of data requested by the number held in the pre-fetch factor register 15. The amount of data to be read depends upon the type of read request as well as the particular system design, for example the size of a

cache line. Table 1 shows the data amounts for three types of read requests. PFFR is the pre-fetch factor register value.

Memory Operation	Alignment	Read Size
Read	DWORD	(PPFR+1)*4*DWORD
Read Line	Cacheline	(PPFR+1)*cacheline
Read multiple	2 cachelines	(PPFR+1)*2 cachelines

A cacheline is a series of contiguous bytes of data corresponding to the host CPU's cache subsystem. Cachelines conform to CPU dependent address alignment. A DWORD is a double word, with a length that depends upon the particular computer memory configuration. Read operations may be limited to cacheline boundaries. Factor is the value contained in the pre-fetch factor register, and may be altered during operation of the computer by software.

Referring to FIG 2, a flow chart of an adaptive read pre-fetch method 100 is shown. At system initialization 105, an initial value for the pre-fetch factor register is set. This may be in system ROM, or may be set (and changed from time to time) as a parameter by the operating system or any other system or application software. In one embodiment, pre-fetch timer may be initialized to a set time, which will decrement to zero unless reset.

If an agent gives a pre-fetchable read request 110 (of whatever type) then the read amount, based upon the type of read, (see table 1) is multiplied by the pre-fetch factor plus one, the pre-fetch factor being stored in the pre-fetch factor register 15 . Thus, if the value of the pre-fetch

factor register is zero, the read amount is multiplied by one, effectively disabling the feature.

The value in the next read register 30 is compared to the value of the read address received from the agent. If they are the same (meaning that the value in the next read address was stored as a result of a prior read request from the same agent which was terminated early for some reason, such as being disconnected by the bridge for lack of data), then the read amount is again increased. The read amount is multiplied by one plus the value in the re-read pre-fetch factor register 20. Other implementations could successively automatically increase the value in the re-read pre-fetch factor register for each early terminated read, and conversely could periodically decrement the re-read pre-fetch factor.

If the address in the read request does not match the value in the next-read address 125, the value in the re-read pre-fetch register is ignored. In either case, the calculated pre-fetch amount is attempted to be read 135.

Table 2 shows the read size for different memory operations using the re-read pre-fetch register (RRPFR) value:

Memory Operation	Alignment	Read Size
Read	DWORD	(PPFR+1+RRPFR)*4*DWORD
Read Line	cacheline	(PPFR+1+RRPFR)*cacheline
Read multiple	2 cachelines	(PPFR+1+RRPFR)*2cachelines

Table 2

If the read terminates early, then the requesting agent has not received all of the data that presumably it presumably wants. Early termination occurs if the bridge disconnects the read transaction because data is exhausted and the requesting device is still expecting additional data (i.e. still

asserting the PCI bus signal FRAME#.) Data may become exhausted because of a variety of reasons, including an end of file, exhaustion of a buffer or other causes.

In the case of a first early termination, the adaptive read pre-fetch process increases the amount of data retrieved on the next read request at the same location (where the current read ended) from the requesting agent. This is accomplished by saving the next-read address (the next address at which data would have been retrieved had the read not been terminated early) and beginning to use the re-read factor and command type specific pre-fetch amounts.

The smart pre-fetch ability may be disabled by programming that is accessible during system initialization and by the operating system as a parameter. A separate process may be implemented for each agent on a secondary bus and may also be implemented in the primary bus side as well as the secondary bus.

The invention has been described in terms of particular embodiments. Other embodiments are within the scope of the following claims. For example, the process may be implemented on a bridge, a separate circuit (discrete or integrated) or in software, or in combinations of software and firmware or circuitry. It may be used successfully in other than a PCI 2.2 bus system. Not all parts of the described embodiment need be implemented to achieve beneficial results.

What is claimed is: